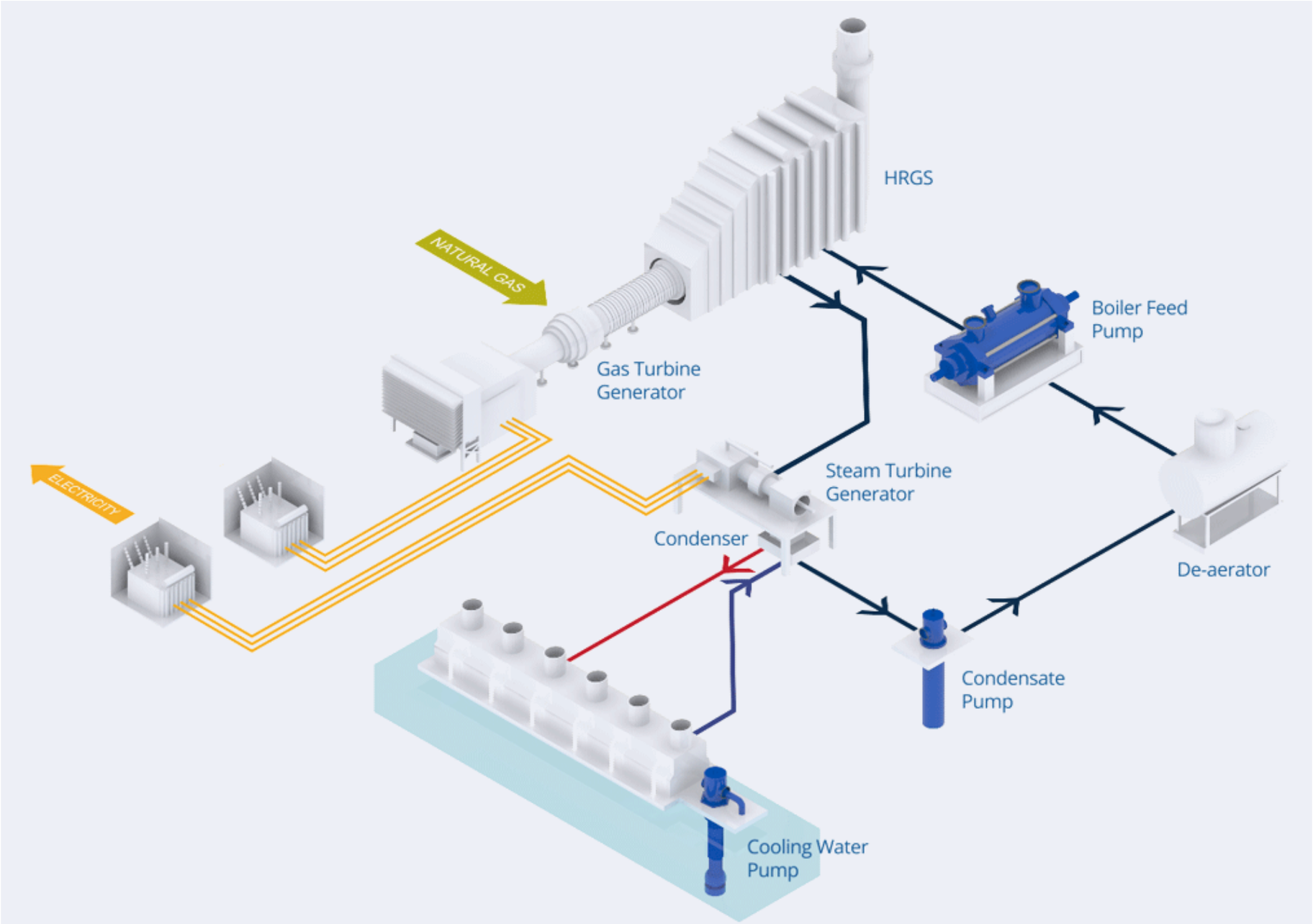
COMBINED CYCLE POWER PLANT DATASET

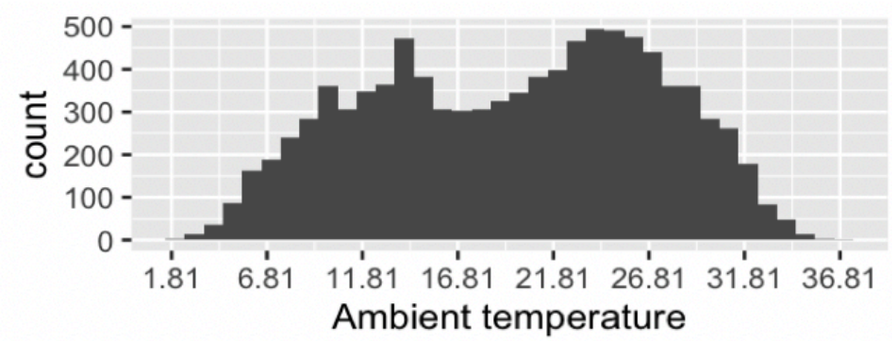
SERGIO ARNAUD

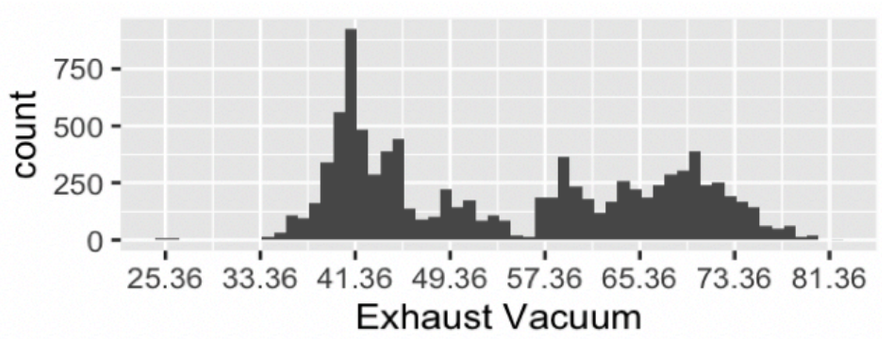
Combined Cycle power plant dataset

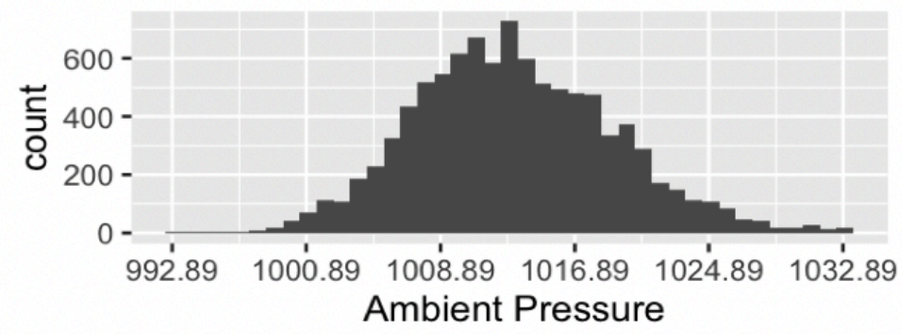


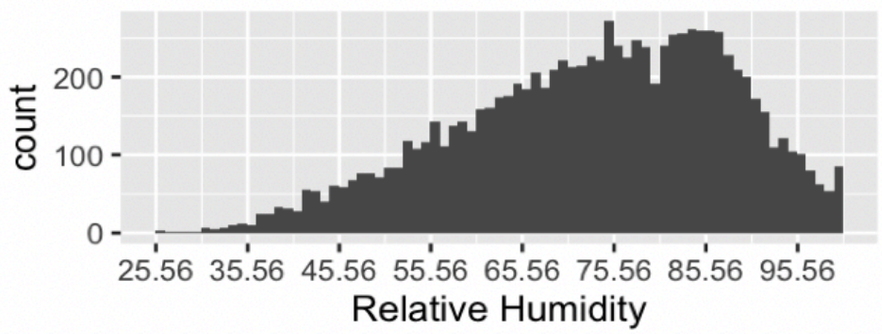
Features

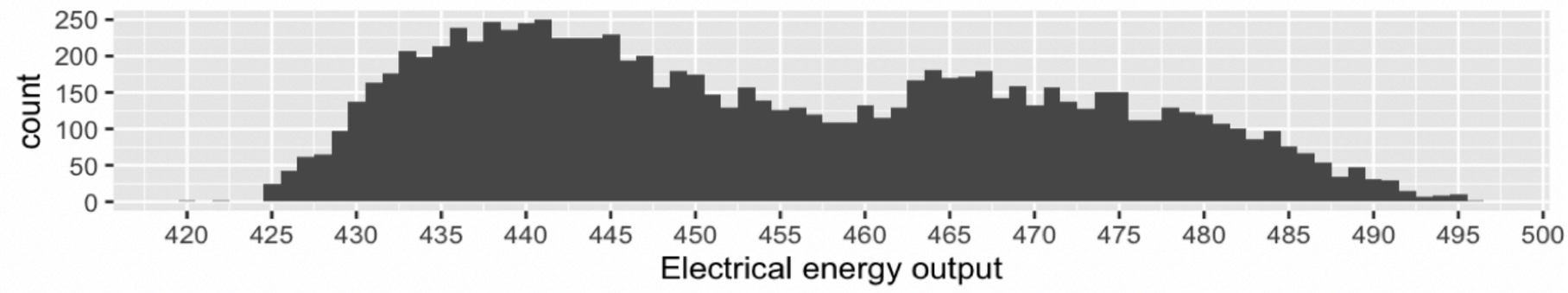
The dataset contains 9568 observations collected from a CCPP over the period 2006-2011 and the features are:.



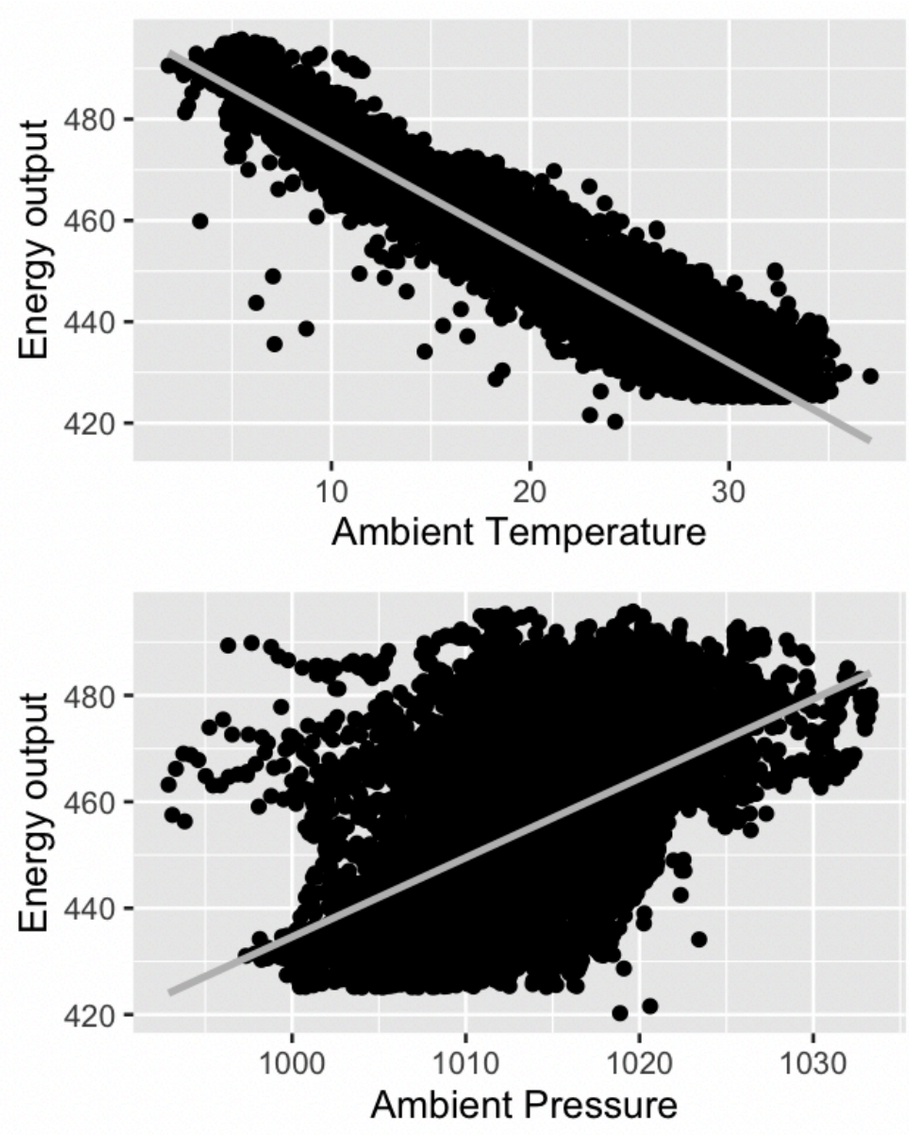


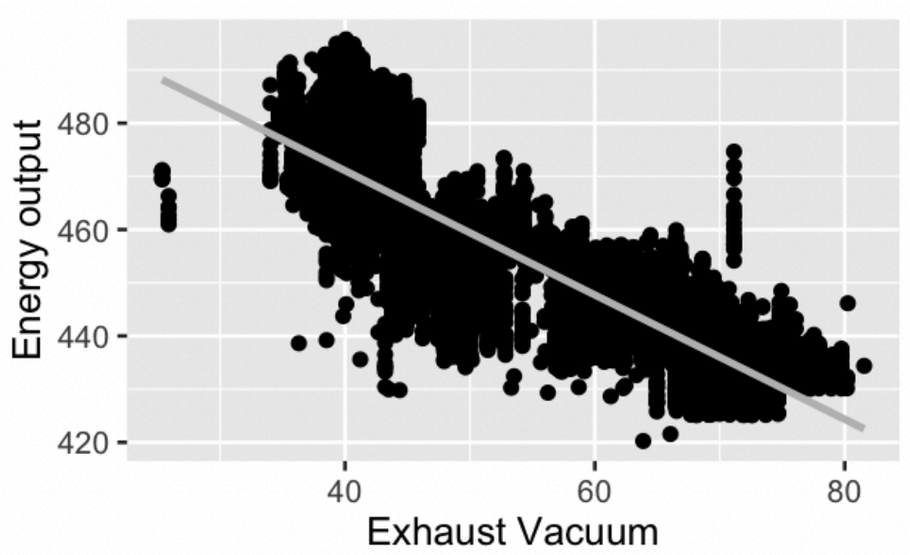


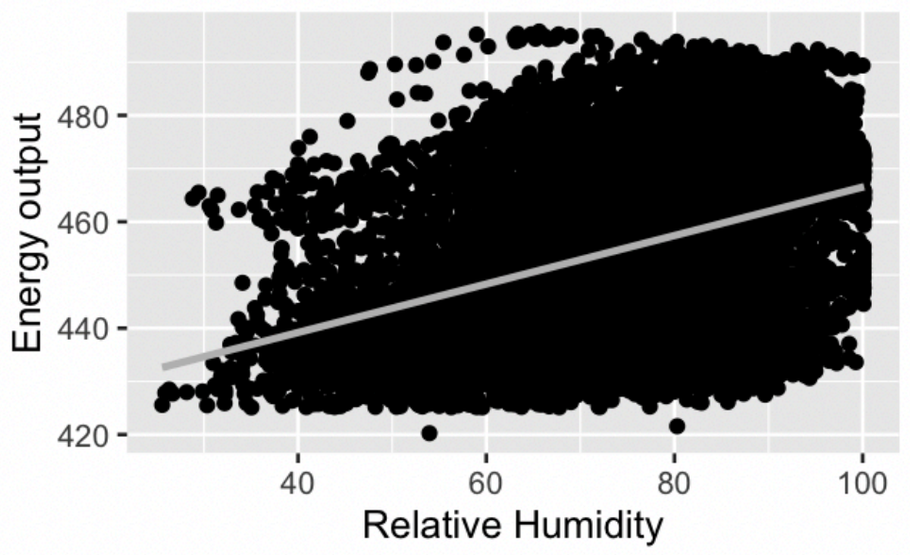




Relations of the predictors with the response







Modelling

Regression task since all the predictors and the response are continuous variables.

A **gradient boosted decision tree** was used to solve the regression task.

A more simple and interpretable model (**linear regression**) was used as a baseline to compare the performance of the gradient boosting model.

Lasso and ridge methods were applied, multicollinearity wasn't a problem and the improvement in terms of test error was minimal so I decided in favor of the more simple linear regression.

Software

Data manipulation and visualization

XGBoost

Optimized distributed library for a gradient boosting framework

Caret package.

Uniform interface to standardize common modelling tasks. In particular for cross-validation

xgboostExplainer package.

Tools to interpret the results of XGBoost

Linear model

All the predictors were significant to the response and a significant regression equation was found:

$$F\,(4, 7651) = 2.517 \times 10^{-4},\ p < 2.2 \times 10^{-16}$$

with

$$R^2 = 0.9294.$$

Gradient boosting (parameter tuning)

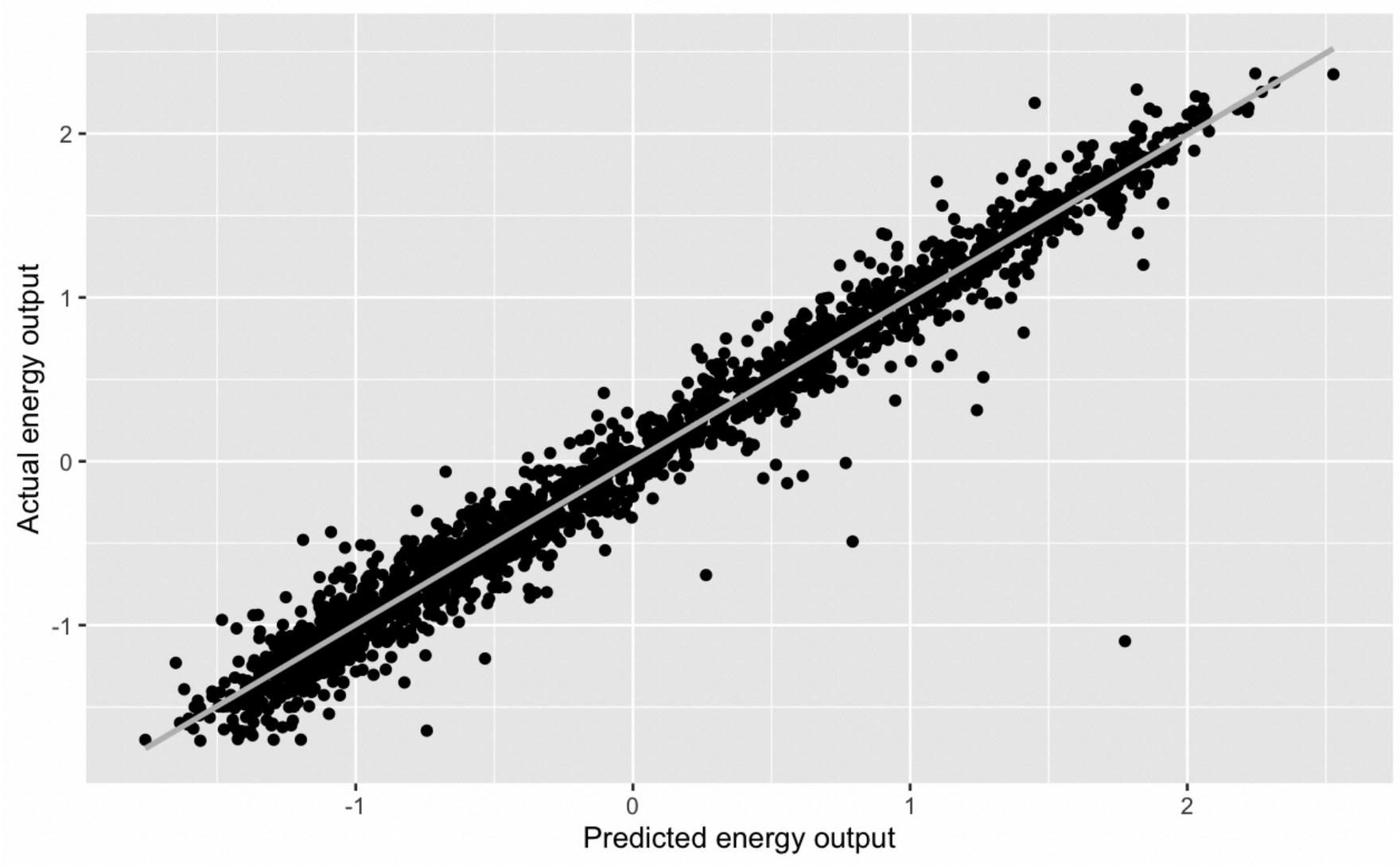
According to XGBoost there are several parameters to be tuned. The most important are:

Parameter	Description	Value
eta	After each boosting step, eta shrinks the feature weights to make the boosting process more conservative and less likely to overfit	.1
gamma	Minimum loss reduction required to make a further partition on a leaf node of the tree. A larger gamma implies a more conservative algorithm.	0
max_depth	Maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit.	6
min_child _weight	If the tree partition step results in a leaf node with the sum of instance weight less than min_child_weight, then the building process will give up further partitioning.	2

Chosen by a grid search and 5 fold cross-validation

Gradient boosting (model evaluation)

The actual vs predicted electrical energy output plot is:



Test mean squared error of 0.032.

Gradient boosting (feature importance)

After the boosted trees are constructed, we can build an importance score for each attribute. That score indicates how valuable each feature was in the fitting process.

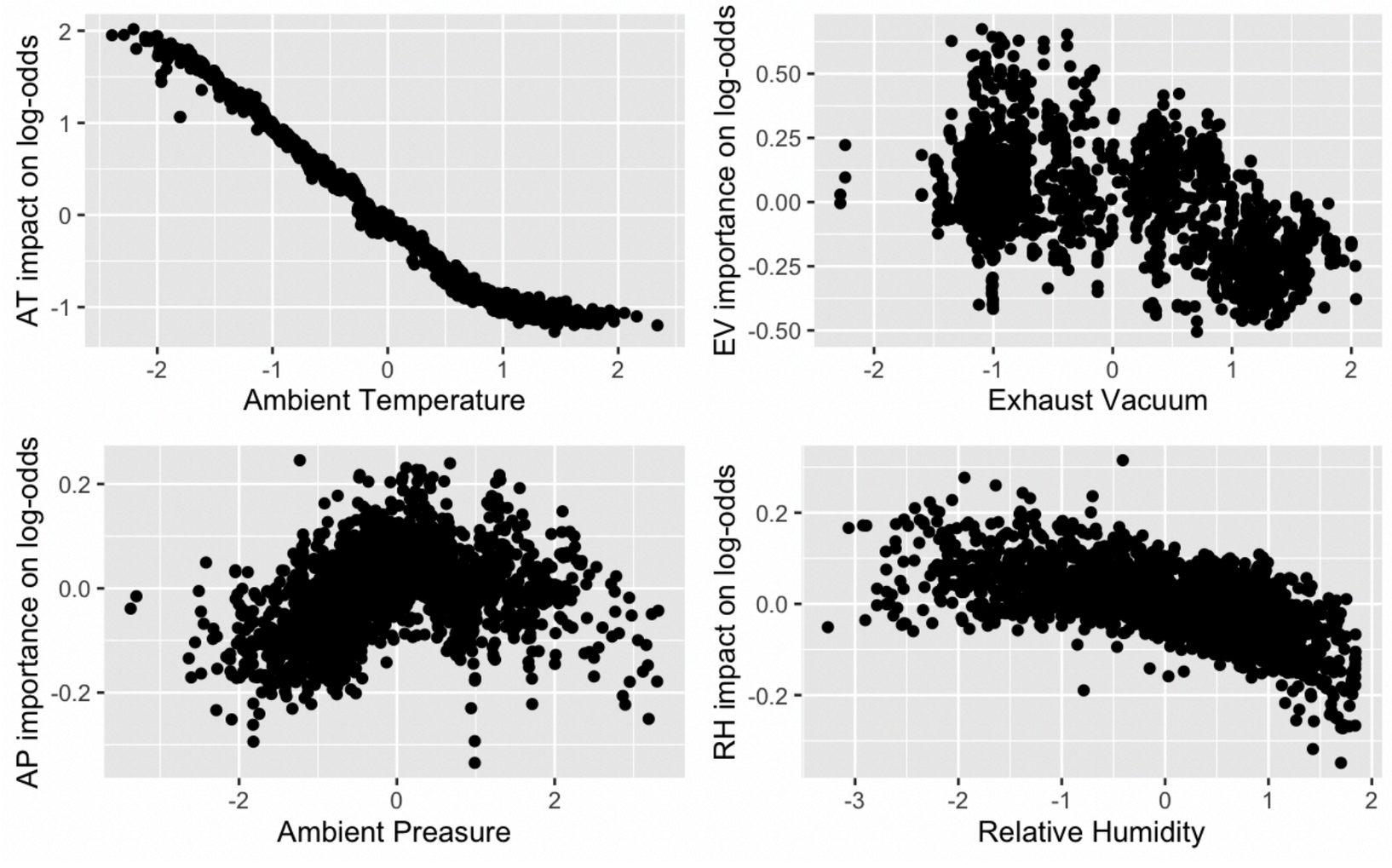
In this case, the feature importance is

Library	Description
Ambient temperature	.90
Exhaust Vacuum	.06
Ambient Preassure	.015
Relative Humidity	.013

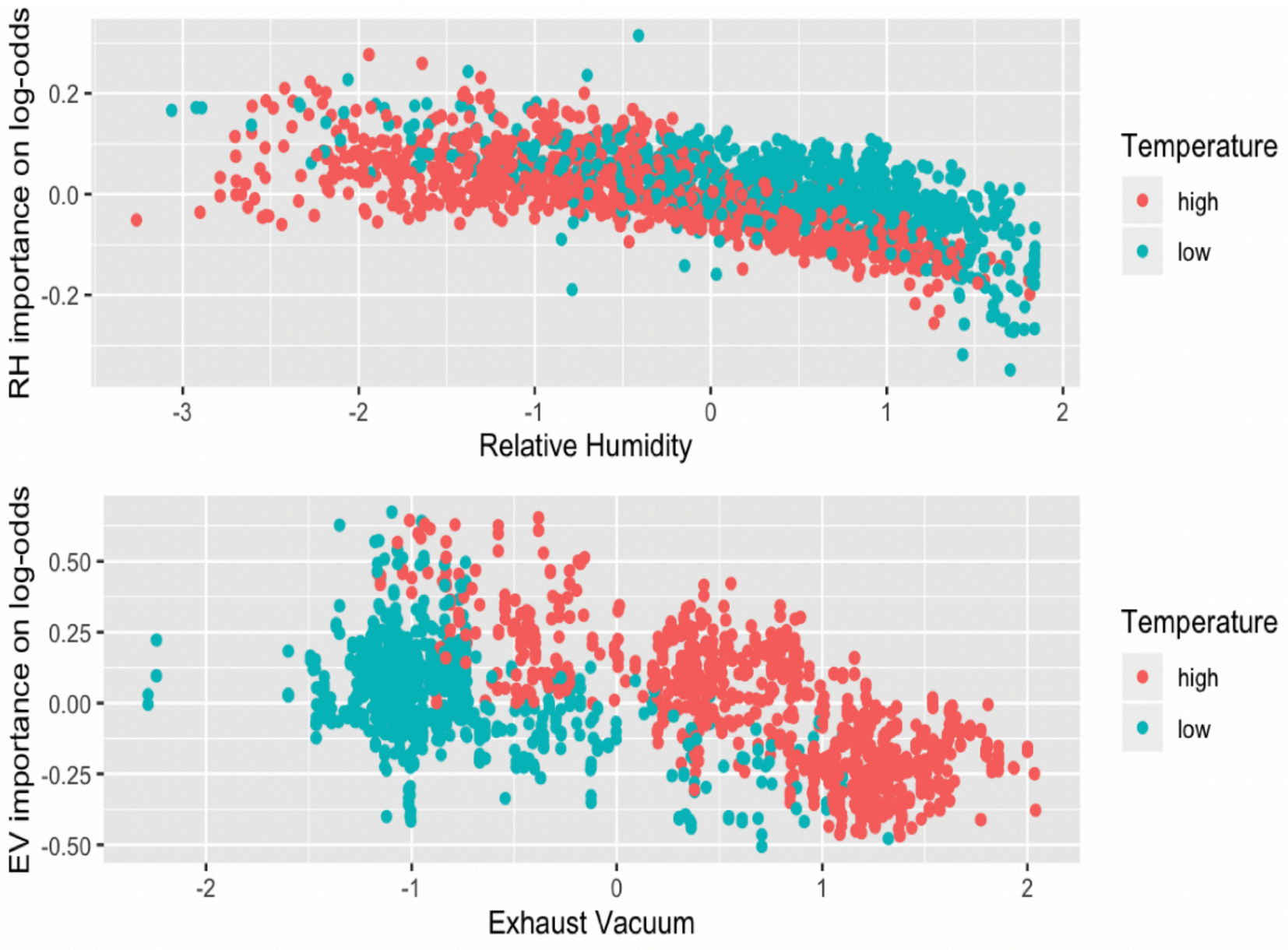
Gradient boosting (Interpretation)

We can extract the log-odds contribution of each feature in a prediction by adding up the contributions of each one of them for every tree in the ensemble.

In that way, we can measure the impact that each feature had in a prediction:



Gradient boosting (Interpretation)



Conclusions

The **linear model** achieved good results and is the most interpretable model.

The **gradient boosted decision tree** model was able to capture some nonlinear relationships between the predictors achieving a mean squared error 76% smaller than the one from the linear model.

The gradient boosting model achieves good prediction power without renouncing to interpretation.

Posible improvements:

- Result interpretation: further analysis + domain knowledge.
- Parameter running: further hyperparameter optimization.
- Modelling: explore other regression techniques
- Dataset: more features